Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 262 (2010) 231–248

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Coalition Description Logic with Individuals

İnanç Seylan[1]

KRDB Research Centre, Free University of Bozen-Bolzano

Wojciech Jamroga[2]

Individual and Collective Reasoning Group, University of Luxembourg

Abstract

Coalitional Description Logic [17] is a product style combination of Coalition Logic and the description logic $\mathcal{ALC}$. It enables reasoning about agents' ability to influence first-order structures. In this paper, we extend the logic with nominals, i.e., names of individuals including those of agents to define more complex terms for coalitions. This extended language allows one to express surprisingly sophisticated properties because it provides a way to reason about how agents can influence themselves. After introducing the new logic formally, we show that its satisfiability problem is still decidable in NEXPTIME. We prove it constructively by proposing a tableau.

Keywords: strategic logic, description logic, multi-agent systems, tableau

1 Introduction

Coalition logic (**CL**) [14,15] formalizes the ability of groups of agents to achieve certain outcomes in strategic games. The central operator of the logic is $[A]$, with $[A]\varphi$ meaning that group of agents A has a strategy to achieve an outcome state where φ holds. On the other hand, description logics (DLs) are logical formalisms for representing the knowledge of an application domain in a structured way [2]. More precisely, DLs allow to describe classes, assign individuals to these classes, and define binary relations on individuals. The importance of DLs lies in the fact that they are decidable fragments of first-order logic and they have well developed practical decision procedures. Moreover, they comprise the formal basis of the Semantic Web ontology languages [10].

[1] Email: seylan@inf.unibz.it

[2] Email: wojtek.jamroga@uni.lu

doi:10.1016/j.entcs.2010.04.017

In [17], we proposed a product style combination of the description logic $\mathcal{ALC}$ with coalition logic, that allowed for application of modal operators to both formulas and concepts. Still, the combination kept the agent and the concept layers pretty much separated; in particular, one could not use the first-order elements of DL to specify how agents and their groups can influence *themselves*. In this paper, we make the first step to overcome the drawback: we extend the language of concepts with names of individuals (including agents), and we allow for more complex terms to define coalitions. This simple extension allows to express surprisingly sophisticated properties, as the examples in Section 2.4 demonstrate. Furthermore, we extend the satisfiability procedure from [17] to handle the new language, and we establish complexity bounds for the satisfiability problem.

It is worth noting that we do not presuppose any general relationship between agents' first-order properties and their temporal abilities in our logic. In particular, if a formula is satisfiable (resp. valid) then one can replace any occurrence of an agent name in the formula with any other agent name, and the resulting formula will be still satisfiable (resp. valid). Note that this symmetry of agents' abilities with respect to the actual actor is a general feature of **CL**, and not specifically of our combination of **CL** and description logic. On the other hand, agents used in coalitions and agents used as individuals *are* semantically related: given a particular model (or a particular theory), we cannot shuffle agent names without changing the truth values of formulae anymore.

2 The Logic

In this section we define our logic formally. We begin by disentangling the syntax and semantics of coalitional expressions: in all the existing literature on strategic logics, no difference is made between sets of "real" agents and the descriptions that refer to them (cf. [15,1] and many others). Instead, it is assumed that the agents themselves occur in formulae of the logics. Such an abuse of formal notation seems acceptable when we refer to agents only in the scope of strategic operators (to name *who* is supposed to achieve the property in question). In our case, however, we want also to reason about agents (resp. coalitions) in the same way as about other individuals (resp. concepts). That is, we want to enjoy the benefits of first-order reasoning with respect to agents and their sets, and make sure that we do it correctly.

After defining the syntax of coalitional expressions (and formulas in which those can occur), we extend our semantics from [17] to handle the new, more sophisticated concepts and agent expressions. Finally, we present some intuitive examples, and show that every formula of $\mathbf{CL}_{\mathcal{ALCO}}$ can be equivalently transformed to a more rigid form, which will prove convenient when defining our decision procedure in further sections.

2.1 Syntax

In order to specify properties of agents and coalitions, we assume the following sets of names: a countable set N_C of *concept names* that includes at least $\mathbb{A}\mathrm{gt}$ (the name for the "grand coalition" of agents), a countable set N_R of *role names*, a finite nonempty set N_I of *individual names*, and a finite nonempty set $N_A \subseteq N_I$ of *agent names*. The set of *concepts* is the smallest set satisfying the following conditions:

- $\top$ is a concept (top concept), and every concept name is a concept;
- If $\mathbf{i}_1, \ldots, \mathbf{i}_n$, $n \geq 0$, are individual names then $\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}$ is a concept (enumeration of individuals);
- If C is a concept and R is a role name then $\forall R.C$ and $\exists R.C$ are concepts;
- If C and D are concepts then $C \sqcap D$, $C \sqcup D$, and $C \backslash\backslash D$ are also concepts;
- If C is a concept and $\mathbb{A}$ is a coalitional term (defined further) then $[\mathbb{A}]C$ and $\langle\mathbb{A}\rangle C$ are concepts.

A *coalitional term* is a concept that includes only names from $N_A \cup \{\mathbb{A}\mathrm{gt}\}$. Therefore, coalitional terms contain no concept names (except $\mathbb{A}\mathrm{gt}$), no role names, and no 'non-agent' individual names.

Additionally, we define $\bot \equiv \{\!|\}\!\}$ (bottom concept), and $\neg C \equiv \top \backslash\backslash C$. Concept names, $\top$, and concepts of type $\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}$ form the set of *atomic concepts*. Also, for any individual names $\mathbf{i}_1, \ldots, \mathbf{i}_n \in N_I$, we will use the following notation: $\mathit{enumterm}(\{\mathbf{i}_1, \ldots, \mathbf{i}_n\}) = \{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}$, and conversely, $\mathit{enumset}(\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}) = \{\mathbf{i}_1, \ldots, \mathbf{i}_n\}$. When we give the semantics it will be clear that no matter how we choose to enumerate a given set of individual names by $\mathit{enumterm}()$, they will yield equivalent interpretations.

Now we can define the set formulas of $\mathbf{CL}_{\mathcal{ALCO}}$ as follows: if C, D are concepts then $C \sqsubseteq D$ and $C = D$ are (atomic) formulas; if φ, ψ are formulas then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi$ are formulas; if φ is a formula and $\mathbb{A}$ is a coalitional term then $[\mathbb{A}]\varphi, \langle\mathbb{A}\rangle\varphi$ are also formulas.[3]

Thus, $\mathbf{CL}_{\mathcal{ALCO}}$ extends the description logic $\mathcal{ALCO}$ with modal operators $[\mathbb{A}], \langle\mathbb{A}\rangle$ for reasoning about how agents can transform the world, but also with means to single out agents from concepts (agent names, special concept $\mathbb{A}\mathrm{gt}$), and with concepts $[\mathbb{A}]C, \langle\mathbb{A}\rangle C$ that group objects depending on how they can be transformed. On the other hand, $\mathbf{CL}_{\mathcal{ALCO}}$ extends coalition logic with elements of dyadic first-order logic, typical for description logics.

2.2 Models

A *model for* $\mathbf{CL}_{\mathcal{ALCO}}$ is a quadruple of the form $\mathfrak{M} = \langle \mathrm{Agt}, W, E, \mathcal{I}\rangle$, where Agt is a finite nonempty set of *agents*, W is a nonempty set of *possible worlds* (*states*), and $E, \mathcal{I}$ associate a *playable effectivity function* E_w and an $\mathcal{ALCO}$*-interpretation* $\mathcal{I}(w)$ with every world $w \in W$.

An effectivity function is defined as $E_w : 2^{\mathrm{Agt}} \to 2^{2^W}$, i.e., a function that assigns

[3] We will sometimes use $[\mathbf{i}_1, \ldots, \mathbf{i}_n]$ (resp. $\langle\mathbf{i}_1, \ldots, \mathbf{i}_n\rangle$) as a shorthand for $[\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}]$ (resp. $\langle\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}\rangle$), both in strategic operators and concept constructors.

a set of achievable *outcomes* $V \subseteq W$ to each coalition $A \subseteq \text{Agt}$. The complement sets for outcomes and coalitions are defined as $\overline{V} = W \setminus V$ and $\overline{A} = \text{Agt} \setminus A$, respectively. E_w is *playable* iff it satisfies the following conditions:

(i) *Semi-seriality*: $\emptyset \notin E_w(A)$, for all coalitions $A \neq \text{Agt}$;
(ii) *Semi-W-completeness*: $W \in E_w(A)$, for all coalitions $A \neq \text{Agt}$;
(iii) *Semi-outcome-monotonicity*: for all $V \subseteq U \subseteq W$ and for all $A \neq \text{Agt}$, if $V \in E_w(A)$ then $U \in E_w(A)$;
(iv) *Semi-superadditivity*: for all V, U, A_1, and A_2 such that $A_1 \cap A_2 = \emptyset$ and $A_1 \cup A_2 \neq \text{Agt}$, if $V \in E_w(A_1)$ and $U \in E_w(A_2)$ then $V \cap U \in E_w(A_1 \cup A_2)$;
(v) Agt-*maximality*: for all V, if $\overline{V} \notin E_w(\emptyset)$ then $V \in E_w(\text{Agt})$;
(vi) *Regularity*: for all V, A, if $V \in E_w(A)$ then $\overline{V} \notin E_w(\overline{A})$.

Additionally, we call an effectivity function *semi-playable* iff it satisfies conditions (i)-(iv). Although the standard definition of playability does not make any exceptions for the grand coalition, we do not lose anything by the above definition of playability because semi-playability together with Agt-maximality and regularity is equivalent to (standard) playability [15].

An $\mathcal{ALCO}$-interpretation $\mathcal{I}(w) = \langle \Delta^{\mathcal{I}(w)}, \cdot^{\mathcal{I}(w)} \rangle$, includes a nonempty set $\Delta^{\mathcal{I}(w)}$ called the *domain* of state w, and a mapping $\cdot^{\mathcal{I}(w)}$ that assigns each concept name C with a subset $C^{\mathcal{I}(w)}$ of $\Delta^{\mathcal{I}(w)}$, each role name R with a binary relation $R^{\mathcal{I}(w)}$ on $\Delta^{\mathcal{I}(w)}$, and each individual name $\mathbf{i}$ with an element $i = \mathbf{i}^{\mathcal{I}(w)}$ in $\Delta^{\mathcal{I}(w)}$. We make the following semantic assumptions wrt $\mathcal{ALCO}$-interpretations:

- *Constant domain*: $\Delta^{\mathcal{I}(w)} = \Delta^{\mathcal{I}(v)}$ for any $w, v \in W$;
- *Global individual names*: $\mathbf{i}^{\mathcal{I}(w)} = \mathbf{i}^{\mathcal{I}(v)}$ for all $\mathbf{i} \in N_I$ and $w, v \in W$;
- *Unique individual names*: $\mathbf{i}_1^{\mathcal{I}(w)} \neq \mathbf{i}_2^{\mathcal{I}(w)}$ for two distinct individual names $\mathbf{i}_1, \mathbf{i}_2 \in N_I$ and all $w \in W$;
- *Correct interpretation of agent names*: $\mathbb{A}\text{gt}^{\mathcal{I}(w)} = \text{Agt}$ and $\{i \mid i = \mathbf{a}^{\mathcal{I}(w)}$ for some $\mathbf{a} \in N_A\} = \text{Agt}$ for every $w \in W$.

As a consequence of our assumptions, agents are a part of domain in every state, and can be referred to like all other concepts and individuals. Moreover, the interpretation of coalitional terms does not change from state to state, and the cardinality of the set of agents Agt must be the same as the number of agent names given in N_A. Note that we have chosen to make the unique individual names assumption because otherwise two different agent names inside modal operators might denote the same agent, leading to a peculiar effect.

In short, models of $\mathbf{CL}_{\mathcal{ALCO}}$ combine information about possible states of reality (W), and how they can be transformed (E), and by whom (Agt), with first-order structures that characterize each possible world separately.

2.3 Semantics

The interpretation $\mathcal{I}(w)$ defines the denotation of individual names and primitive concepts in state w. We extend it to concept descriptions in a similar way to [17]:

$$
\begin{aligned}
\top^{\mathcal{I}(w)} &= \Delta^{\mathcal{I}(w)},\\
\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\}^{\mathcal{I}(w)} &= \{\mathbf{i}_1^{\mathcal{I}(w)}, \ldots, \mathbf{i}_n^{\mathcal{I}(w)}\},\\
(C \sqcap D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \cap D^{\mathcal{I}(w)},\\
(C \sqcup D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \cup D^{\mathcal{I}(w)},\\
(C \backslash\backslash D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \setminus D^{\mathcal{I}(w)},\\
(\forall R.C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \forall \delta' \, (\langle \delta, \delta' \rangle \in R^{\mathcal{I}(w)} \rightarrow \delta' \in C^{\mathcal{I}(w)})\},\\
(\exists R.C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \exists \delta' \, (\langle \delta, \delta' \rangle \in R^{\mathcal{I}(w)} \wedge \delta' \in C^{\mathcal{I}(w)})\},\\
([\mathbb{A}]C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid \|C\|_{\delta}^{\mathfrak{M}} \in E_w(\mathbb{A}^{\mathcal{I}(w)})\},\\
(\langle \mathbb{A} \rangle C)^{\mathcal{I}(w)} &= \{\delta \in \Delta^{\mathcal{I}(w)} \mid W \setminus \|C\|_{\delta}^{\mathfrak{M}} \notin E_w(\mathbb{A}^{\mathcal{I}(w)})\},
\end{aligned}
$$

where $\|C\|_{\delta}^{\mathfrak{M}} = \{w \in W \mid \delta \in C^{\mathcal{I}(w)}\}$ is the set of states that δ belongs to the interpretation of concept C.

Now we define the satisfaction relation $\models$ for $\mathbf{CL}_{\mathcal{ALCO}}$ as follows:

$$
\begin{aligned}
&\mathfrak{M}, w \models C \sqsubseteq D \text{ iff } C^{\mathcal{I}(w)} \subseteq D^{\mathcal{I}(w)},\\
&\mathfrak{M}, w \models C = D \text{ iff } C^{\mathcal{I}(w)} = D^{\mathcal{I}(w)},\\
&\mathfrak{M}, w \models \neg\varphi \quad \text{iff } \mathfrak{M}, w \not\models \varphi,\\
&\mathfrak{M}, w \models \varphi \wedge \psi \quad \text{iff } \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi,\\
&\mathfrak{M}, w \models \varphi \vee \psi \quad \text{iff } \mathfrak{M}, w \models \varphi \text{ or } \mathfrak{M}, w \models \psi,\\
&\mathfrak{M}, w \models [\mathbb{A}]\varphi \quad \text{iff } \|\varphi\|^{\mathfrak{M}} \in E_w(\mathbb{A}^{\mathcal{I}(w)}),\\
&\mathfrak{M}, w \models \langle \mathbb{A} \rangle \varphi \quad \text{iff } W \setminus \|\varphi\|^{\mathfrak{M}} \notin E_w(\mathbb{A}^{\mathcal{I}(w)}),
\end{aligned}
$$

where $\|\varphi\|^{\mathfrak{M}} = \{w \in W \mid \mathfrak{M}, w \models \varphi\}$ is the set of states that satisfy φ in $\mathfrak{M}$. We observe that the possibility/necessity of φ being true in the next moment can be expressed with operators $[\mathbb{A}\mathrm{gt}], [\,]$: $[\mathbb{A}\mathrm{gt}]\varphi$ can be read as "there is a possible next state for which φ holds", while $[\,]\varphi$ expresses that in every possible next state φ is the case.

A formula φ is *satisfiable* if there exist a model $\mathfrak{M} = \langle W, E, \mathcal{I} \rangle$ and a state $w \in W$ such that $\mathfrak{M}, w \models \varphi$. A concept C is *satisfiable* if there exist $\mathfrak{M} = \langle W, E, \mathcal{I} \rangle$ and $w \in W$ such that $C^{\mathcal{I}(w)} \neq \emptyset$. Concept D *subsumes* concept C if $C^{\mathcal{I}(w)} \subseteq D^{\mathcal{I}(w)}$ for all models $\mathfrak{M} = \langle W, E, \mathcal{I} \rangle$ and all $w \in W$. Note that concept subsumption and concept satisfiability can be reduced to formula (un)satisfiability. Concept C is satisfiable iff formula $\neg(C \sqsubseteq \bot)$ is satisfiable and concept D subsumes concept C iff formula $\neg(C \sqsubseteq D)$ is unsatisfiable. The formula $C \sqsubseteq D$ is clearly equivalent to $\top \sqsubseteq \neg C \sqcup D$, and $C = D$ to $\top \sqsubseteq (\neg C \sqcup D) \sqcap (\neg D \sqcup C)$. In the remainder of this paper, we will assume without loss of generality that every atomic formula is of the form $\top \sqsubseteq E$ and we will restrict our attention to satisfiability of formulas.

2.4 Examples, Properties, Remarks

In the examples, we will use $\mathbf{a}_1, \mathbf{a}_2, \ldots$ for agents' names, and $a_1, a_2, \ldots$ for the agents that these names refer to.

Example 2.1 Formula $[\mathbf{a}](\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{Happy})$ specifies that agent a can make himself happy. We can write similar specifications for coalitions: $[\mathbf{a}_1, \mathbf{a}_2](\{\!|\mathbf{a}_1{,}\mathbf{a}_2|\!\} \sqsubseteq \mathit{Happy})$ says that a_1 and a_2 can make themselves happy if they cooperate. Furthermore, $[\mathbf{a}_1, \mathbf{a}_2](\mathbb{A}\mathrm{gt} \sqsubseteq \mathit{Happy})$ states that a_1, a_2 can make all the agents happy.

Consider formula $(\mathbb{A}\mathrm{gt} \sqcap [\mathbf{a}]\mathit{Happy}) \sqsubseteq \mathit{Happy}$. Clearly, it says that a cannot make anybody *happier* than now. Moreover, $\langle \mathbb{A}\mathrm{gt} \backslash\backslash \{\!|\mathbf{a}|\!\} \rangle \big((\mathbb{A}\mathrm{gt} \sqcap [\mathbf{a}]\mathit{Happy}) \sqsubseteq \mathit{Happy} \big)$ adds that all the other agents can do nothing to change this sad state of affairs.

Example 2.2 Let Perm stand for the set of permissions to be in a building, and In represent the set of agents that are currently inside. Formula $\bigwedge_{\mathbf{a} \in N_A} \big([\mathbf{admin}](\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{Perm}) \wedge [\mathbf{admin}] \neg (\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{Perm}) \big)$ specifies that the administrator can grant and deny the permission to any agent. Moreover, $\bigwedge_{\mathbf{a} \in N_A} \neg (\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{In}) \rightarrow \big(\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{Perm} \leftrightarrow [\mathbf{a}](\{\!|\mathbf{a}|\!\} \sqsubseteq \mathit{In}) \big)$ says that an agent is able to enter the building if, and only if, he has a permission to do so.

Note that the specifications refer only to the agents' abilities *in the current moment*, while it would be rather more appropriate to specify them as invariants of the scenario. This is one of the drawbacks of coalition logic, and a reason why extending our logic with **ATL** operators [1] seems an interesting avenue for future research.

Example 2.3 Consider a system with a dynamic hierarchy of processes captured by the *parent* role name, and a dynamic configuration of active processes represented by the concept *Active*. Formula $\bigwedge_{\mathbf{a} \in N_A} (\mathit{Active} \sqcap [\mathbf{a}] \neg \mathit{Active}) = \exists \mathit{parent}.\{\!|\mathbf{a}|\!\}$ says that agents can deactivate exactly those processes they are parents of. Adding a requirement that an agent cannot activate a process without becoming its parent: $\bigwedge_{\mathbf{a} \in N_A} \neg \mathit{Active} \sqcap [\mathbf{a}](\mathit{Active} \sqcap \neg \exists \mathit{parent}.\{\!|\mathbf{a}|\!\}) = \bot$ makes for a viable specification of a hierarchic multi-process system.

In the next sections we will present a procedure that constructs models for such specifications (provided they are satisfiable). Before we go on to presenting our satisfiability algorithm, however, we show that the input formula of the procedure can be given in a slightly simplified form without any loss of generality.

Lemma 2.4 *Let $\mathbb{A}, \mathbb{B}$ be coalitional terms. Every coalitional term in the form $[\mathbb{A}]\mathbb{B}$ is equivalent to $\mathbb{B}$, i.e., $([\mathbb{A}]\mathbb{B})^{\mathcal{I}(w)} = \mathbb{B}^{\mathcal{I}(w)}$ for every model $\mathfrak{M}$ and state w. The same holds for $\langle \mathbb{A} \rangle \mathbb{B}$.*

Proof. $([\mathbb{A}]\mathbb{B})^{\mathcal{I}(w)} = \{\delta \in \Delta^{\mathcal{I}(w)} \mid \|\mathbb{B}\|^{\mathfrak{M}}_{\delta} \in E_w(\mathbb{A}^{\mathcal{I}(w)})\}$ where $\|\mathbb{B}\|^{\mathfrak{M}}_{\delta} = \{w' \in W \mid \delta \in \mathbb{B}^{\mathcal{I}(w')}\}$. Since the interpretation of coalitional terms does not change from state to state, we get that $\|\mathbb{B}\|^{\mathfrak{M}}_{\delta} = W$ if $\delta \in \mathbb{B}^{\mathcal{I}(w)}$ and $\emptyset$ otherwise. By playability of E_w, we have $W \in E_w$ and $\emptyset \notin E_w$, so $([\mathbb{A}]\mathbb{B})^{\mathcal{I}(w)} = \{\delta \in \Delta^{\mathcal{I}(w)} \mid \delta \in \mathbb{B}^{\mathcal{I}(w)}\} = \mathbb{B}^{\mathcal{I}(w)}$.

The proof for $\langle \mathbb{A} \rangle \mathbb{B}$ is analogous. □

A formula is *singleton reduced* iff every enumeration of individuals not appearing inside a modal operator is of the form $\{\!|\mathbf{i}|\!\}$; *difference free* iff it does not contain $\backslash\backslash$; in *negation normal form* (NNF) iff negation signs appear only in front of atomic formulas, concept names, and enumeration of individuals; and *coalitionally simple* iff all the coalitional terms it includes are of the form $\{\!|\mathbf{a}_1, \ldots, \mathbf{a}_k|\!\}$, with $\mathbf{a}_1, \ldots, \mathbf{a}_k \in N_A$.

A formula can be transformed to an equivalent singleton reduced formula by the concept equivalences $\{\!|\mathbf{i}_1, \ldots, \mathbf{i}_n|\!\} \equiv \{\!|\mathbf{i}_1|\!\} \sqcup \ldots \sqcup \{\!|\mathbf{i}_n|\!\}$ [2] and $\{\!|\,|\!\} \equiv \bot$; to a difference free formula by the equivalence $C \backslash\backslash D \equiv C \sqcap \neg D$; and to a formula in NNF by making use of de Morgan's laws, the duality between value restrictions and full existential quantifications, and the duality between modal operators. We denote the NNF of formula φ (resp. concept C) by $\dot{\neg}\varphi$ (resp. $\dot{\neg}C$). For coalitionally simple formulas, we use the following proposition.

Proposition 2.5 *For every formula φ there is an equivalent coalitionally simple formula φ', i.e., such that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}, w \models \varphi'$ for every $\mathfrak{M}, w$. Moreover, φ' is at most linearly longer than φ, and can be obtained in linear time wrt the length of φ and the number of agent names in N_A. More precisely, $|\varphi'| = O(|\varphi| \cdot |N_A|)$, and it can be obtained in time $O(|\varphi| \cdot |N_A|)$.*

Proof. We use the following translation scheme:

- tr_1 replaces every occurrence of $\mathbb{A}\mathrm{gt}$ in φ with $enumterm(N_A)$;
- tr_2 replaces every occurrence of $[\mathbb{A}]\mathbb{B}$ and $\langle\mathbb{A}\rangle\mathbb{B}$ with $\mathbb{B}$ (recursively, proceeding from atomic concepts and subformulas to more complex ones);
- tr_3 replaces (recursively) every occurrence of $\{\!|\mathbf{a}_1, \ldots, \mathbf{a}_n|\!\} \sqcap \{\!|\mathbf{a}'_1, \ldots, \mathbf{a}'_m|\!\}$ (where $\mathbf{a}_i, \mathbf{a}'_i \in N_A$) with $enumterm(\{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \cap \{\mathbf{a}'_1, \ldots, \mathbf{a}'_m\})$, and analogously for $\sqcup, \backslash\backslash$.

Then, $\varphi' = tr_3(tr_2(tr_1(\varphi)))$ is coalitionally simple and equivalent to φ (the proof is straightforward). □

In fact, we will mainly consider coalitionally simple, singleton reduced, and difference free formulas in negation normal form.

Remark 2.6 We have chosen to interpret the set N_A as including the names of *existing* rather than *potential* agents. It may seem somewhat restrictive when writing specifications: after we have decided on a precise syntax (including the set of agent names N_A), the number of agents is completely fixed in the models of our logic. Alternatively, we could assume a countable set of *available* agent names, and provide denotations for only a finite subset of those in each particular model. We are planning to explore the possibility in the future.

3 Hintikka Structures for $\mathrm{CL}_{\mathcal{ALCO}}$

Decision procedures based on semantic tableaux construct not a model of the given formula, but a structure closely resembling a model, called a *Hintikka structure* [16]. A Hintikka structure makes use of Hintikka sets [9] which are also called downward

saturated sets in the literature. Some authors prefer using alternative names such as *model graph* [7] or *tableau* [11] for referring to the same abstraction of a model. In this section, we specify Hintikka structures for $\mathbf{CL}_{\mathcal{ALCO}}$-formulas. The proofs showing their equivalence to $\mathbf{CL}_{\mathcal{ALCO}}$ models are omitted for reasons of space but they combine results of [17,11].

For a $\mathbf{CL}_{\mathcal{ALCO}}$-formula φ, denote by

- $\mathsf{con}(\varphi)$ the set of all $\mathbf{CL}_{\mathcal{ALCO}}$-concepts occurring in φ,
- $\mathsf{rol}(\varphi)$ the set of all role names occurring in φ,
- $\mathsf{for}(\varphi)$ the set of all subformulas of φ,
- $\mathsf{con}^{\dot{\neg}}(\varphi) = \mathsf{con}(\varphi) \cup \{\dot{\neg}C \mid C \in \mathsf{con}(\varphi)\}$,
- $\mathsf{fcl}(\varphi) = \mathsf{for}(\varphi) \cup \{[\,]\vartheta \mid \langle enumterm(N_A)\rangle\vartheta \in \mathsf{for}(\varphi)\}$,
- $\mathsf{ccl}(\varphi) = \mathsf{con}^{\dot{\neg}}(\varphi) \cup \{[\,]C \mid \langle enumterm(N_A)\rangle C \in \mathsf{con}^{\dot{\neg}}(\varphi)\} \cup \{\{\!|\mathbf{i}|\!\} \mid \mathbf{i} \in N_I\}$.

Definition 3.1 If φ is a $\mathbf{CL}_{\mathcal{ALCO}}$ formula, a *basic Hintikka structure* for φ is defined to be a hextuple $H = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\rangle$ such that

- Σ is a nonempty set of states,
- $\Lambda : \Sigma \to 2^{\mathsf{fcl}(\varphi)}$ maps each state to a set of formulas which is a subset of $\mathsf{fcl}(\varphi)$,
- $\mathbf{S}$ is a map associating with each $w \in \Sigma$ a nonempty set of concept types,
- $\mathcal{L}$ associates with each state $w \in \Sigma$ a function
$$\mathcal{L}_w : \mathbf{S}(w) \to 2^{\mathsf{ccl}(\varphi)}$$
that maps each concept type s in $\mathbf{S}(w)$ to a set of concepts which is a subset of $\mathsf{ccl}(\varphi)$,
- $\mathcal{E}$ associates with each state $w \in \Sigma$ a function
$$\mathcal{E}_w : \mathsf{rol}(\varphi) \to 2^{\mathbf{S}(w)\times\mathbf{S}(w)}$$
that maps each role R in $\mathsf{rol}(\varphi)$ to a set of pairs of concept types from $\mathbf{S}(w)$,
- $\mathcal{J}$ associates with each state $w \in \Sigma$ a function
$$\mathcal{J}_w : N_I \to \mathbf{S}(w)$$
that maps individual names to concept types in $\mathbf{S}(w)$ such that $\mathbf{i} \neq \mathbf{j}$ implies $\mathcal{J}_w(\mathbf{i}) \neq \mathcal{J}_w(\mathbf{j})$, and $\mathcal{J}_w(\mathbf{i}) = \mathcal{J}_v(\mathbf{i})$ for all $v \in \Sigma$,
- there is some $w_\varphi \in \Sigma$ such that $\varphi \in \Lambda(w_\varphi)$.

Furthermore, for all $w \in \Sigma$, $s,t \in \mathbf{S}(w)$, $\vartheta, \vartheta_1, \vartheta_2 \in \mathsf{fcl}(\varphi)$, $C, C_1, C_2 \in \mathsf{ccl}(\varphi)$, $R \in \mathsf{rol}(\varphi)$, $\mathbf{i} \in N_I$, it holds that:

($\mathsf{P}_\bot$) $\bot \notin \mathcal{L}_w(s)$;
($\mathsf{P}_\neg$) if $C \in \mathcal{L}_w(s)$ then $\neg C \notin \mathcal{L}_w(s)$, where C is a concept name or of the form $\{\!|\mathbf{i}|\!\}$;
($\mathsf{P}_\sqcap$) if $C_1 \sqcap C_2 \in \mathcal{L}_w(s)$ then $C_1 \in \mathcal{L}_w(s)$ and $C_2 \in \mathcal{L}_w(s)$;
($\mathsf{P}_\sqcup$) if $C_1 \sqcup C_2 \in \mathcal{L}_w(s)$ then $C_1 \in \mathcal{L}_w(s)$ or $C_2 \in \mathcal{L}_w(s)$;
($\mathsf{P}_\forall$) if $\forall R.C \in \mathcal{L}_w(s)$ and $\langle s,t\rangle \in \mathcal{E}_w(R)$ then $C \in \mathcal{L}_w(t)$;
($\mathsf{P}_\exists$) if $\exists R.C \in \mathcal{L}_w(s)$ then there is some $s' \in \mathbf{S}$ such that $\langle s,s'\rangle \in \mathcal{E}_w(R)$ and $C \in \mathcal{L}_w(s')$;
($\mathsf{P}_\mathbf{i}$) $\{\!|\mathbf{i}|\!\} \in \mathcal{L}_w(s)$ iff $s = \mathcal{J}(\mathbf{i})$;
($\mathsf{P}_\sqsubseteq$) if $\top \sqsubseteq C \in \Lambda(w)$ then $C \in \mathcal{L}_w(s)$;
($\mathsf{P}_{\not\sqsubseteq}$) if $\neg(\top \sqsubseteq C) \in \Lambda(w)$ then there is some $s' \in \mathbf{S}$ such that $\dot{\neg}C \in \mathcal{L}_w(s')$;

($\mathsf{P}_\wedge$) if $\vartheta_1 \wedge \vartheta_2 \in \Lambda(w)$ then $\vartheta_1 \in \Lambda(w)$ and $\vartheta_2 \in \Lambda(w)$;
($\mathsf{P}_\vee$) if $\vartheta_1 \vee \vartheta_2 \in \Lambda(w)$ then $\vartheta_1 \in \Lambda(w)$ or $\vartheta_2 \in \Lambda(w)$;
(P_{N_A}) if $\langle enumterm(N_A)\rangle C \in \mathcal{L}_w(s)$ then $[\,]C \in \mathcal{L}_w(s)$, and
if $\langle enumterm(N_A)\rangle \vartheta \in \Lambda(w)$ then $[\,]\vartheta \in \Lambda(w)$.

Modal operators, regardless of being in front of formulas or concepts, impose additional properties on basic Hintikka structures. To define these properties in a more uniform way, we will use a notational convenience.

Definition 3.2 Let $\langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\rangle$ be a basic Hintikka structure for φ. For a state $w \in \Sigma$, the set Φ_w is defined as

$$\Phi_w := \Lambda(w) \cup \{s : C \mid C \in \mathcal{L}_w(s) \text{ and } s \in \mathbf{S}\}.$$

α and β are placeholders for elements of Φ_w. $\bigtriangledown\alpha$ is either equal to some $\bigtriangledown\vartheta$ or $s : \bigtriangledown C$, where $\bigtriangledown \in \{[\mathbb{A}], \langle\mathbb{A}\rangle\}$ for some coalitional term $\mathbb{A}$. If $\bigtriangledown\alpha = \bigtriangledown\vartheta$ then $\alpha = \vartheta$; and if $\bigtriangledown\alpha = s : \bigtriangledown C$ then $\alpha = s : C$.

As the reader familiar with tableau would notice, the meanings of symbols α and β in our uniform notation are different than in Smullyan's notation to classify formulas [4]. We are interested in certain set of modal expressions called modal saturations.

Definition 3.3 $\Psi \subseteq \Phi_w$ is called a *modal saturation* in a state $w \in \Sigma$ of a basic Hintikka structure if and only if Ψ is equal to

($\mathsf{S}_{[\mathbb{A}]}$) $\{[\mathbb{A}_1]\alpha_1, \ldots, [\mathbb{A}_n]\alpha_n\}$ such that $\mathbf{a} \in enumset(\mathbb{A}_i) \cap enumset(\mathbb{A}_j)$ implies $i = j$,
($\mathsf{S}_{[\mathbb{A}\rangle}$) $\{\langle\mathbb{A}\rangle\alpha, [\mathbb{A}_1]\alpha_1, \ldots, [\mathbb{A}_n]\alpha_n\}$ such that $\mathbf{a} \in enumset(\mathbb{A}_i) \cap enumset(\mathbb{A}_j)$ implies $i = j$ and $\bigcup_{i=1}^{n} enumset(\mathbb{A}_i) \subseteq enumset(\mathbb{A})$, or
($\mathsf{S}_{\langle\mathbb{A}\rangle}$) $\{\langle\mathbb{A}\rangle\alpha\}$.

The first tableau based decision procedure for **CL** is developed by Hansen [8]. Her formulation of modal saturations is simpler than ours because she instead chooses to have a separate property corresponding to superadditivity. However, such a property generates new conjunctions which are not subformulas of the original formula. In particular to our case where we also deal with concepts, this would require us to define a conjunction operator over $\bigtriangledown\vartheta$ and $s : \bigtriangledown C$ which we wished to avoid.

We are now in a position to define a Hintikka structure which is essentially equivalent to a $\mathbf{CL}_{\mathcal{ALCO}}$-model as far as the satisfiability of a $\mathbf{CL}_{\mathcal{ALCO}}$-formula is concerned.

Definition 3.4 Let $H = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J}\rangle$ be a basic Hintikka structure for φ. H is said to be a *Hintikka structure* for φ if and only if:

(P_{CD}) $\mathbf{S}(w) = \mathbf{S}(v)$, for all $w, v \in \Sigma$;
($\mathsf{P}^H_{\mathsf{S}}$) for all $w \in \Sigma$, if $\{\bigtriangledown_1\alpha_1, \ldots, \bigtriangledown_n\alpha_n\}$ is a modal saturation in w then there is some $v \in \Sigma$ such that $\alpha_1, \ldots, \alpha_n \in \Phi_v$.

Lemma 3.5 *A* $\mathbf{CL}_{\mathcal{ALCO}}$*-formula* φ *is satisfiable iff there exists a Hintikka structure for* φ*.*

Representing individuals[4] explicitly in tableau algorithms for expressive fragments of first-order modal logics with constant domains is problematic as far as termination is concerned. This observation is first made for modal description logics by Baader and Laux [3]. Later, quasimodels are proposed to show the decidability of various modal description logics [18]. Only after then it was possible to devise tableau based decision procedures for these logics [13,12] because the algorithms relied on the finite representation of individuals as offered by quasimodels. To this end, an individual (type) is defined as a function over the set of states.

Definition 3.6 Let $H = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J} \rangle$ be a basic Hintikka structure. A *run* r in H is a function associating with every $w \in \Sigma$ a concept type $r(w)$ in $\mathbf{S}(w)$.

A run r is defined over all states so that an individual (in the classical sense) corresponding to r is represented by some type in every state. This validates the constant domain assumption. However, one needs to impose several properties on runs in order for them to preserve satisfiability.

Definition 3.7 Let $H = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J} \rangle$ be a basic Hintikka structure for φ. A *Hintikka quasistructure* for φ is a tuple $Q = \langle H, \mathbf{R} \rangle$, where $\mathbf{R}$ is a set of runs in H. Furthermore, it holds that:

($\mathsf{P}^{\mathbf{i}}_r$) for every $\mathbf{i}$ in N_I, the run $r_{\mathbf{i}}$ defined by $r_{\mathbf{i}}(w) = \mathcal{J}_w(\mathbf{i})$, for all $w \in \Sigma$, is in $\mathbf{R}$;
(P^{s}_r) for every $w \in \Sigma$ and every s in $\mathbf{S}(w)$ such that $s \notin \mathsf{codom}(\mathcal{J}_w)$, there exists a run r in $\mathbf{R}$ such that $r(w) = s$;
($\mathsf{P}^{Q}_{\mathsf{S}}$) for every $w \in \Sigma$ and every $r \in \mathbf{R}$, if Ψ is a modal saturation in w then there is some v in Σ such that $r(w) : \bigtriangledown C \in \Psi$ implies $r(v) : C \in \Phi_v$ and $\bigtriangledown \vartheta \in \Psi$ implies $\vartheta \in \Phi_v$.

Lemma 3.8 *Let* φ *be a* $\mathbf{CL}_{\mathcal{ALCO}}$*-formula. There exists a Hintikka quasistructure for* φ *iff there exists a Hintikka structure for* φ*.*

A more compact representation of a Hintikka quasistructure is possible by relaxing the definition of a run so that it can associate more than a single type for a state. However, we won't define these 'overloaded' runs but dissolve their effect into a global property in the resulting structures.

Definition 3.9 A basic Hintikka structure $\langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J} \rangle$ for φ is a *compact Hintikka quasistructure* for φ if and only if for every $w \in \Sigma$,

($\mathsf{P}^{C}_{\mathsf{S}}$) if Ψ is a modal saturation in w then there is some v in Σ such that
- for every $\mathbf{i}$ in N_I, $\mathcal{J}_w(\mathbf{i}) : \bigtriangledown C \in \Psi$ implies $\mathcal{J}_v(\mathbf{i}) : C \in \Phi_v$;
- $s : \bigtriangledown C \in \Psi$ and $s \notin \mathsf{codom}(\mathcal{J}_w)$ implies there exists a type t in $\mathbf{S}(v)$ such that $\{C \mid s : \bigtriangledown C \in \Psi\} \subseteq \mathcal{L}_w(t)$;
- $\bigtriangledown \vartheta \in \Psi$ implies $\vartheta \in \Phi_v$.

[4] types in the context of our Hintikka structures

Example 3.10 Consider the compact Hintikka quasistructure $H = \langle \Sigma, \Lambda, \mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{J} \rangle$ for φ with $\Sigma = \{w, v\}$ and $\mathcal{L}_w(s) = \{[\mathbf{a}]C, [\mathbf{b}]D, [\mathbf{b}, \mathbf{c}]E\}$, $\mathcal{L}_v(s) = \{C, D\}$, $\mathcal{L}_v(t) = \{C, E\}$ (it does not matter how φ actually looks like). H is not a Hintikka quasistructure for φ because there exists no run r in H with $r(w) = s$, and hence, (P^s_r) is violated. The proof is easy, if there were to be an r with $r(w) = s$, whatever choice we make for $r(v)$, i.e., $r(v) = s$ or $r(v) = t$, $(\mathsf{P}^Q_{\mathsf{S}})$ would be violated. However, it is possible to modify H and convert it into a Hintikka quasistructure by duplicating the state v with all the necessary mappings.

The proof of the following Lemma generalizes the observation we made in our example.

Lemma 3.11 *Let φ be a $\mathbf{CL}_{\mathcal{ALCO}}$-formula. There exists a compact Hintikka quasistructure for φ iff there exists a Hintikka quasistructure for φ.*

Theorem 3.12 *A $\mathbf{CL}_{\mathcal{ALCO}}$ formula φ is satisfiable iff there exists a compact Hinttika quasistructure for φ.*

Proof. Follows immediately from Lemmas 3.5, 3.8, and 3.11. □

4 Tableau Algorithm for $\mathbf{CL}_{\mathcal{ALCO}}$

From Theorem 3.12, an algorithm which constructs a (finite) representation of a compact Hintikka quasistructure for a $\mathbf{CL}_{\mathcal{ALCO}}$-formula can be used as a decision procedure for the satisfiability of $\mathbf{CL}_{\mathcal{ALCO}}$-formulas. In this section, we describe such an algorithm, and we prove its termination, soundness, and completeness.

4.1 Definition of the Algorithm

Let N_V be a set of countably infinite variables, and $<$ be the well-order relation on N_V. A *term* is either a variable or an individual name. Let φ be a $\mathbf{CL}_{\mathcal{ALCO}}$ formula. A *constraint* for φ is (i) a formula in $\mathsf{fcl}(\varphi)$, (ii) an atom of the form $x : C$ where x is a term and $C \in \mathsf{ccl}(\varphi)$, or (iii) an atom of the form $(x, y) : R$ where x, y are terms and $R \in \mathsf{rol}(\varphi)$. A *constraint system* S for φ is a finite, nonempty set of constraints for φ. A *completion set* $\mathbf{T}$ for φ is a set of constraint systems for φ.

In order to avoid superfluous definitions, we will abuse the α, β uniform notation that we defined in Definition 3.2 for Hintikka structures. To this end, α and β are placeholders for constraints. $\bigtriangledown\alpha$ is either equal to some $\bigtriangledown\vartheta$ or $x : \bigtriangledown C$, where $\bigtriangledown \in \{[\mathbb{A}], \langle\mathbb{A}\rangle\}$ for some coalitional term $\mathbb{A}$ and x is a term. If $\bigtriangledown\alpha = \bigtriangledown\vartheta$ then $\alpha = \vartheta$; and if $\bigtriangledown\alpha = x : \bigtriangledown C$ then $\alpha = x : C$. Constraints of type $\bigtriangledown\alpha$ are also called *modal constraints*. More precisely, $[\mathbb{A}]\alpha$ is referred to as a *positive modal constraint* and $\langle\mathbb{A}\rangle\alpha$ as a *negative modal constraint*. Furthermore, a negative modal constraint $\langle\mathbb{A}\rangle\alpha$ is called proper if $\mathit{enumset}(\mathbb{A}) \neq N_A$.

A term x is in S if S contains a mention of x. A variable x is *fresh* for S if x is not in S and $x > y$ for all variables y in S. We denote by $S[x/\mathbf{i}]$ the constraint system obtained from S by substituting every occurrence of the variable x with the

The $\mathsf{R}_{\wedge}$ rule	
Condition:	$\vartheta_1 \wedge \vartheta_2 \in S$ and $\{\vartheta_1, \vartheta_2\} \not\subseteq S$.
Action:	$S := S \cup \{\vartheta_1, \vartheta_2\}$.
The $\mathsf{R}_{\vee}$ rule	
Condition:	$\vartheta_1 \vee \vartheta_2 \in S$ and $\{\vartheta_1, \vartheta_2\} \cap S = \emptyset$.
Action:	$S := S \cup \{\psi\}$ for some $\psi \in \{\vartheta_1, \vartheta_2\}$.
The $\mathsf{R}_{\sqcap}$ rule	
Condition:	$x : C_1 \sqcap C_2 \in S$ and $\{x : C_1, x : C_2\} \not\subseteq S$.
Action:	$S := S \cup \{x : C_1, x : C_2\}$.
The $\mathsf{R}_{\sqcup}$ rule	
Condition:	$x : C_1 \sqcup C_2 \in S$ and $\{x : C_1, x : C_2\} \cap S = \emptyset$.
Action:	$S := S \cup \{x : E\}$ for some $E \in \{C_1, C_2\}$.
The $\mathsf{R}_{\exists}$ rule	
Condition:	$x : \exists R.C \in S$, x is not blocked w.r.t. S, and x has no R-successor y in S with $y : C \in S$.
Action:	$S := S \cup \{(x, y) : R, y : C\}$, where y is fresh for S.
The $\mathsf{R}_{\forall}$ rule	
Condition:	$x : \forall R.C \in S$, there is a R-successor y of x in S with $y : C \notin S$.
Action:	$S := S \cup \{y : C\}$.
The $\mathsf{R}_{\mathbf{i}}$ rule	
Condition:	$x : \{\!|\mathbf{i}|\!\} \in S$ for x a variable in S.
Action:	$S := S[x/\mathbf{i}]$.
The $\mathsf{R}_{\sqsubseteq}$ rule	
Condition:	$\top \sqsubseteq C \in S$ and $x : C \notin S$ for a term x in S.
Action:	$S := S \cup \{x : C\}$.
The $\mathsf{R}_{\not\sqsubseteq}$ rule	
Condition:	$\neg(\top \sqsubseteq C) \in S$ and there is no term x such that $x : \dot{\neg}C \in S$.
Action:	$S := S \cup \{x : \dot{\neg}C\}$, where x is fresh for S.
The R_{N_A} rule	
Condition:	$\langle enumterm(N_A)\rangle\alpha \in S$ and $[\,]\alpha \notin S$.
Action:	$S := S \cup \{[\,]\alpha\}$.

Fig. 1. Local expansion rules for $\mathbf{CL}_{\mathcal{ALCO}}$.

individual name $\mathbf{i}$. If $(x, y) : R \in S$ for some role name R and terms x, y then y is called a *R-successor* of x in S, or just a successor when R is not important.

A variable x is *blocked* in S if there is some other variable y such that $\{C \mid x : C \in S\} \subseteq \{D \mid y : D \in S\}$ and $y < x$. In this case, we say that y *blocks* x in S. S (and therefore $\mathbf{T}$ if $S \in \mathbf{T}$) is said to contain a *clash* if

- $x : \bot \in S$ for some term x,
- $\{x : C, x : \neg C\} \subseteq S$ for some term x and some concept name C,
- $\mathbf{i} : \neg\{\!|\mathbf{i}|\!\} \in S$ for some individual name $\mathbf{i}$, or
- $\mathbf{i} : \{\!|\mathbf{j}|\!\} \in S$ for individiual names $\mathbf{i}, \mathbf{j}$.

The *local expansion rules* are given in Figure 1. The definition of the *global expansion rule* which adds new constraint systems into the completion set is more involved. In what follows, we will introduce this rule. We say that a rule, regardless of its type, is *applicable* to a constraint system S iff its condition is satisfied in S.

Let S be a constraint system. The equivalence relation $\sim_S$ on the set of variables occurring in S is defined by taking $x \sim_S y$ iff $\{C \mid x : C \in S\} = \{D \mid y : D \in S\}$. The equivalence class generated by x is denoted by $[x]_S$. Finally, $\sim(S) = \{min([x]_S) : C \mid x : C \in S\} \cup \{\vartheta \mid \vartheta \in S\}$.

If S and S' are constraint systems then S' is called a *variant* of S, written $S \approx S'$, iff there is a bijective function f from the variables in S onto the variables in S' such that S' is obtained from S by replacing each variable x from S with $f(x)$.

Analogous to the modal saturations defined in Definition 3.3 for Hintikka structures, a subset S' of a constraint system S is called a *modal saturation in* S if and only if S' is equal to

(i) $\{[\mathbb{A}_1]\alpha_1, \ldots, [\mathbb{A}_n]\alpha_n\}$ such that $\mathbf{a} \in \mathit{enumset}(\mathbb{A}_i) \cap \mathit{enumset}(\mathbb{A}_j)$ implies $i = j$,

(ii) $\{\langle\mathbb{A}\rangle\alpha, [\mathbb{A}_1]\alpha_1, \ldots, [\mathbb{A}_n]\alpha_n\}$ such that $\mathbf{a} \in \mathit{enumset}(\mathbb{A}_i) \cap \mathit{enumset}(\mathbb{A}_j)$ implies $i = j$ and $\bigcup_{i=1}^{n} \mathit{enumset}(\mathbb{A}_i) \subseteq \mathit{enumset}(\mathbb{A})$, or

(iii) $\{\langle\mathbb{A}\rangle\alpha\}$,

such that none of the modal constraints mention a blocked variable.

The global expansion rule $\mathsf{R}_{[\mathbb{A}\rangle}$ is similar to the one in [6] and defined as follows.

Condition: S is not marked as finished.

Action:

(i) Order linearly all positive and proper negative modal constraints in S not mentioning blocked variables in such a way that all the positive modal constraints precede all the proper negative ones. Suppose the result is the following list:
$\mathbb{L} = [\mathbb{A}_0]\alpha_0, \ldots, [\mathbb{A}_{m-1}]\alpha_{m-1}, \langle\mathbb{A}'_0\rangle\beta_0, \ldots, \langle\mathbb{A}'_{n-1}\rangle\beta_{n-1}$.
Let $|\mathbb{L}|$ be the length of $\mathbb{L}$, i.e., $m+n$. Denote by $\Theta(S)$ the set $\{0, \ldots, |\mathbb{L}|\}^{\sharp N_A}$. As in the relational model, we identify the components of a tuple τ in $\Theta(S)$ by names, more precisely agent names, so that $\tau_{\mathbf{a}}$ corresponds to the component of τ identified by $\mathbf{a}$. Lastly, for every $\tau \in \Theta(S)$, denote by $\mathrm{pos}(\tau)$ the set $\{\mathbf{a} \mid \tau_{\mathbf{a}} \geq m\}$; and by $\mathrm{neg}(\tau)$ the number $\left[\sum_{\mathbf{a}\in\mathrm{pos}(\tau)}(\tau_{\mathbf{a}} - m)\right] \bmod n$.

(ii) $\mathsf{res} := \{\}$, where res is a set of constraint systems.

(iii) Consider the elements of $\Theta(S)$ in the lexicographic order and for each $\tau \in \Theta(S)$ do the following:
  (a) Create a constraint system

$$S_\tau = \{\alpha_i \mid [\mathbb{A}_i]\alpha_i \in S \text{ and } \forall \mathbf{a} \in \mathit{enumset}(\mathbb{A}_i).\ \tau_{\mathbf{a}} = i\} \cup \{\beta_j \mid \langle\mathbb{A}'_j\rangle\beta_j \in S,\ \mathrm{neg}(\tau) = j, \text{ and } N_A \setminus \mathit{enumset}(\mathbb{A}'_j) \subseteq \mathrm{pos}(\tau)\}.$$

  (b) If $S_\tau = \emptyset$ then continue with the next iteration.
  (c) $S_\tau := {\sim}(S_\tau \cup \{\mathbf{a} : \{\!|\mathbf{a}|\!\} \mid \mathbf{a} \in N_I\} \cup \{x : \top\})$, where x is fresh for S.
  (d) If $S_\tau \not\approx S'_\tau$, for all $S'_\tau \in \mathsf{res}$, then $\mathsf{res} := \mathsf{res} \cup \{S_\tau\}$.

(iv) $\mathbf{T} := \mathbf{T} \cup \{\mathsf{res}\}$.

(v) Mark S as finished.

Let φ be the $\mathbf{CL}_{\mathcal{ALCO}}$-formula to be tested for satisfiability. The tableau algorithm starts with the *initial completion set* $\mathbf{T} = \{S_0\}$ for φ, where $S_0 = \{\varphi\} \cup \{\mathbf{i} : \{\!|\mathbf{i}|\!\} \mid \mathbf{i} \in N_I\} \cup \{x_0 : \top\}$, x_0 being the first variable from N_V. $\mathbf{T}$ is then expanded by repeatedly applying the rules in such a way that the global expansion rule is applied only when none of the local expansion rules is applicable, and among local expansion rules $\mathsf{R}_\exists$ or $\mathsf{R}_{\not\sqsubseteq}$ is applicable only when none of the other local expansion

rules is applicable to a constraint system. The expansion continues until the resulting completion set contains a clash or none of the rules is applicable to it. Such a completion set is called *complete*. If the expansion rules can be applied to $\mathbf{T}$ in such a way that they yield a complete and clash-free completion set then the algorithm returns "φ is satisfiable", and "φ is unsatisfiable" otherwise. Note that the tableau algorithm is a *nondeterministic algorithm* due to $\mathsf{R}_\vee$ and $\mathsf{R}_\sqcup$. Each of these rules chooses which disjunct to add for a disjunctive formula (concept).

4.2 Correctness and Termination

Theorem 4.1 (Termination) *Let $n = \sharp\mathsf{ccl}(\varphi) + \sharp\mathsf{fcl}(\varphi) + \sharp N_I$, where $\sharp$ denotes set cardinality. When started with the initial completion set $\mathbf{T}$ for φ, the tableau algorithm terminates after the number of steps exponential in n.*

Proof. We first show that a local expansion rule can be applied at most once to the same constraint. If we didn't have the rule $\mathsf{R}_\mathbf{i}$, this is obvious to see. In the presence of $\mathsf{R}_\mathbf{i}$, if $\mathsf{R}_\mathbf{i}$ is applied to some $x : \{|\mathbf{i}|\} \in S$ then $\{\mathbf{i} : C \mid x : C \in S\} \cup \{(y, \mathbf{i}) : R \mid (y, x) : R \in S)\} \subseteq S[x/\mathbf{i}]$ [5]. For a term y of which x is a R-successor, this ensures that neither $\mathsf{R}_\exists$ nor $\mathsf{R}_\forall$ will be applied again to some $y : \exists R.C \in S$ or $y : \forall R.C \in S$, respectively. If $x : \dot{\neg}C \in S$ and $\top \not\sqsubseteq C \in S$ then $\mathbf{i} : \dot{\neg}C \in S[x/\mathbf{i}]$ ensures that $\mathsf{R}_{\not\sqsubseteq}$ will not be applied to $\top \not\sqsubseteq C \in S$. For all the other local expansion rules, the same holds trivially.

There can be at most $2^{\sharp\mathsf{ccl}(\varphi)}$ unblocked variables in S by the definition of blocking. New variables can only be introduced to S by $\mathsf{R}_\exists$ or $\mathsf{R}_{\not\sqsubseteq}$. Call these rules generating and all the other local expansion rules non-generating. If $\mathsf{R}_\exists$ has been applied to a constraint $x : \exists R.C$ or $x : \dot{\neg}C$ has been added to S by $\mathsf{R}_{\not\sqsubseteq}$ then x will never be blocked in S because by the strategy of rule applications S was closed under the application of non-generating rules. Therefore, if a variable x is blocked then it is introduced by $\mathsf{R}_\exists$, x is the successor of some unblocked term y, and there is no successor of x. As a consequence, the number of blocked variables in S can not be more than $2^{\sharp\mathsf{ccl}(\varphi)}$ which is the case when every unblocked variable blocks a variable and all blocked variables are pairwise disjoint.

Obviously, there may exist at most $\sharp\mathsf{ccl}(\varphi)$ constraints of the form $x : C$ for each term x in S. All individual names in N_I are in S. Hence $\sharp\mathsf{ccl}(\varphi) \cdot (\sharp N_I + 2^{\sharp\mathsf{ccl}(\varphi)+1})$ is the upper bound on the number of constraints of the form $x : C$ in S, where x is a term. A constraint of the form $(x, y) : R$ is always introduced with a constraint $x : C$. Therefore, the number of constraints of the form $(x, y) : R$ is limited by the number of constraints of the form $x : C$.

The number of constraints of the form ϑ in S can not exceed $\sharp\mathsf{fcl}(\varphi)$. Combining this with the upper bound on the number of constraints with terms, the cardinality of S is at most $2 \cdot \sharp\mathsf{ccl}(\varphi) \cdot (\sharp N_I + 2^{\sharp\mathsf{ccl}(\varphi)+1}) + \sharp\mathsf{fcl}(\varphi) \leq 2n(n + 2^{n+1}) + n = 2n^2 + n \cdot 2^{n+2} + n = \epsilon$.

All local expansion rules except $\mathsf{R}_\mathbf{i}$ strictly expand S and as we have shown a local expansion rule can be applied at most once to the same constraint. Thus, the

[5] Notice that by the strategy of rule applications, x can never have a successor before $\mathsf{R}_\mathbf{i}$ is applied

number of local expansion rule applications – not counting the ones for $\mathsf{R}_{\mathbf{i}}$ – can be at most ϵ, i.e., the maximal cardinality of S. We need an upper bound on the number of applications of $\mathsf{R}_{\mathbf{i}}$. Observe that in every $x : \{\!|\mathbf{i}|\!\}$ that triggers $\mathsf{R}_{\mathbf{i}}$, x is a variable. This means $\mathsf{R}_{\mathbf{i}}$ can not be applied more than the sum of the times that generating rules are applied. Therefore, it takes no more than $2 \cdot \epsilon$ steps to apply all local expansion rules to S.

Before we determine the number of steps it takes to apply all global completion rules, we need a few definitions. The *modal depth* $\mathrm{md}(\psi)$ of ψ is the length of the longest chain of nested modal operators in ψ (both in subformulas and subconcepts). The modal depth $\mathrm{md}(x : C)$ of a constraint $x : C$ is defined analogously. The modal depth $\mathrm{md}(S)$ of a constraint system S is the maximal modal depth of constraints in S.

The *depth* of a tree is the number of edges in its longest branch; the *outdegree* is the maximal number of immediate successors of nodes in the tree.

$|\mathbb{L}| \leq \sharp S$ and thus, $\sharp\Theta(S) = \epsilon^{\sharp \mathrm{Agt}} \leq \epsilon^n = \rho$. This is also the upper bound on $\sharp\mathsf{res}$. So $\mathsf{R}_{[\mathbb{A}\rangle}$ can add at most ρ new constraint systems to $\mathbf{T}$ at S. Let S' be one of these new constraint systems. Clearly, $\mathrm{md}(S') < \mathrm{md}(S)$. Define a tree T from $\mathbf{T}$ in the following way.

(i) The root of T is S_0.

(ii) If S is a node in T then S' such that S' is generated by $\mathsf{R}_{[\mathbb{A}\rangle}$ from S is a successor of S.

From our discussion it follows immediately that the depth of this tree is $\mathrm{md}(\varphi)$ and the outdegree is ρ. $\mathrm{md}(\varphi) \leq n$, and thus, the number of nodes in T is at most $\rho^0 + \ldots + \rho^n = \zeta$. As $\mathsf{R}_{[\mathbb{A}\rangle}$ is the only rule that expands this tree, the total number of steps it takes to apply all global completion rules is bounded by the same number.

Since we know that there are ζ nodes in T and it takes $2 \cdot \epsilon$ steps to apply all local expansion rules per node, $2 \cdot \epsilon \cdot \zeta$ is the total number of applications of local expansion rules during the run of the algorithm. Thus, the sum $2 \cdot \epsilon \cdot \zeta + \zeta$ is the total number of applications of all expansion rules which is exponential in n. □

Corollary 4.2 *The tableau algorithm runs in* NEXPTIME.

Remark 4.3 Pauly obtains the PSPACE lower bound of the satisfiability problem of **CL** by a polynomial reduction from the satisfiability problem of **KD** (the normal modal logic over serial frames) [15]. The formula satisfiability problem of $\mathbf{KD}_{\mathcal{ALC}}$ is NEXPTIME-hard [5]. Therefore, we conjecture that the NEXPTIME-hardness of $\mathbf{CL}_{\mathcal{ALC}}$ [17] (which is subsumed by $\mathbf{CL}_{\mathcal{ALCO}}$) could be shown by a similar reduction.

The following Lemma will be useful for establishing the soundness and completeness of the tableau algorithm. It says that $\mathsf{R}_{[\mathbb{A}\rangle}$ finds all and only modal saturations in the constraint system to which it is applied.

Lemma 4.4 *Suppose S is a constraint system and that $\mathsf{R}_{[\mathbb{A}\rangle}$ has been applied to S. A subset $\bigtriangledown S'$ of S is a modal saturation in S if and only if there is some $\tau \in \Theta(S)$ such that $S_\tau = S' \neq \emptyset$.*

Proof. Denote by $\bigtriangledown S$ a constraint system containing only modal constraints and let $S = \{\alpha \mid \bigtriangledown\alpha \in \bigtriangledown S\}$. It is enough to consider subsets of S containing only modal constraints because those are the ones actually by $\mathsf{R}_{[\mathbb{A}\rangle}$. Thus, we look at all possible combinations of modal constraints.

(1) Suppose $\bigtriangledown S'$ be a subset of S that consists only of positive modal constraints and let $\mathsf{ind}(\bigtriangledown S')$ be the set of all indices i such that an element of $\bigtriangledown S'$ has the position i in $\mathbb{L}$. Suppose $\bigtriangledown S'$ is a modal saturation in S. Then for all $i, k \in \mathsf{ind}(\bigtriangledown S')$, $\mathbf{a} \in \mathit{enumset}(\mathbb{A}_i) \cap \mathit{enumset}(\mathbb{A}_k)$ implies $i = k$. For every $i \in \mathsf{ind}(\bigtriangledown S')$ and every $\mathbf{a} \in \mathit{enumset}(\mathbb{A}_i)$, set $\tau_{\mathbf{a}} = i$ and for every $\mathbf{a} \in N_A \setminus \bigcup_{i\in\mathsf{ind}(S')} \mathit{enumset}(\mathbb{A}_i)$, set $\tau_{\mathbf{a}} = \mathit{min}(\mathsf{ind}(\bigtriangledown S'))$. By the definition of $\Theta(S)$, τ is in $\Theta(S)$; and $S_\tau = S'$. Now suppose $\bigtriangledown S'$ is not a modal saturation in S. Then for some $i, k \in \mathsf{ind}(\bigtriangledown S')$, there is an $\mathbf{a} \in \mathit{enumset}(\mathbb{A}_i) \cap \mathit{enumset}(\mathbb{A}_k)$. This means if there were to be a $\tau \in \Theta(S)$ such that $S_\tau = S'$ then $\tau_{\mathbf{a}} = i$ and $\tau_{\mathbf{a}} = k$. But this is not possible.

(2) Let $\bigtriangledown S' = \{\langle\mathbb{A}\rangle\alpha\}$. Any such $\bigtriangledown S'$ is a modal saturation in S; therefore, it is enough to show that there is some $\tau \in \Theta(S)$ such that $S_\tau = S' \neq \emptyset$. We consider two cases.

Case $\mathit{enumset}(\mathbb{A}) \neq N_A$: Let k be the position of $\langle\mathbb{A}\rangle\alpha$ in $\mathbb{L}$. By definition, $k \geq m$ and by assumption, there exists $\mathbf{a}' \in N_A \setminus \mathit{enumset}(\mathbb{A})$. Set $\tau_{\mathbf{a}'} = k$ and set $\tau_{\mathbf{a}} = m$ for every $\mathbf{a} \in N_A \setminus \{\mathbf{a}'\}$. As a consequence, $\mathrm{neg}(\tau) = k - m$ and $N_A \setminus \mathit{enumset}(\mathbb{A}) \subseteq \mathrm{pos}(\tau)$ which means that $\{\alpha\} = S_\tau$.

Case $\mathit{enumset}(\mathbb{A}) = N_A$: By R_{N_A}, $[\ ]\alpha$ is in S and moreover it is a modal saturation in S. But by Case 1, we know that there is some $\tau \in \Theta(S)$ such that $S_\tau = \{\alpha\}$.

(3) Let $\bigtriangledown S'$ be a subset of S that contains more than one negative modal constraint. $\bigtriangledown S'$ is not a modal saturation in S and there is no $\tau \in \Theta(S)$ such that $S_\tau = S'$ because $\mathrm{neg}(\tau)$ allows us to choose at most one negative modal constraint.

(4) Let $\bigtriangledown S'$ be a subset of S containing some positive modal constraints and one negative modal constraint. The proof is a combination of the proofs for Cases 1 and 2. □

Corollary 4.5 *Suppose* $\mathbf{T}$ *is a completion set with* $S \in \mathbf{T}$ *and that* $\mathsf{R}_{[\mathbb{A}\rangle}$ *has been applied to* S. *If* S' *is a modal saturation in* S *then there is some* S'' *in* $\mathbf{T}$ *such that*

- *for every* $\mathbf{i}$ *in* N_I, $\mathbf{i} : \bigtriangledown C \in S'$ *implies* $\mathbf{i} : C \in S''$;
- *for a variable* x, $x : \bigtriangledown C \in S'$ *implies there exists a variable* y *in* S'' *such that* $\{C \mid x : \bigtriangledown C \in S'\} \subseteq \{D \mid y : D \in S''\}$;
- $\bigtriangledown\vartheta \in S'$ *implies* $\vartheta \in S''$.

The proofs of following theorems are extensions of the respective ones in [17].

Theorem 4.6 (Soundness) *If the tableau algorithm returns "*φ *is satisfiable" for a* $\mathbf{CL}_{\mathcal{ALCO}}$*-formula* φ *then* φ *is satisfiable.*

Theorem 4.7 (Completeness) *If a* $\mathbf{CL}_{\mathcal{ALCO}}$*-formula* φ *is satisfiable then the tableau algorithm returns "*φ *is satisfiable".*

5 Conclusions

In this paper, we extend Coalition Description Logic [17] with nominals for individuals, and we use these nominals to refer to coalitions. This might be considered as a relatively simple extension of CDL, but it is well-balanced between expressivity and difficulty of reasoning. Regarding expressive power, one can reason about agents' ability to influence themselves and to the best of our knowledge this is the first such coalition logic. As for the difficulty of reasoning, we use only basic set operations to reason in the modal component of the logic. Allowing more concept constructors to define coalitions would require model-checking techniques from the DL component; it looks like an interesting research direction for the future.

This study offers also an interesting reflection on the methodological level. In our previous work [17], we proposed a general framework for combining strategic and descriptive elements of a multi-agent system. However, it turned out that a nontrivial semantic machinery does not guarantee exciting specifications. Here, we present a small extension of the framework (one may be even tempted to call it *slight*) that allows to specify surprisingly sophisticated properties, as we hope to have demonstrated in Section 2.4. This shows that the practical usability of a language may very much depend on elements which seem minor from the theoretical point of view.

References

[1] Alur, R., T. A. Henzinger and O. Kupferman, *Alternating-time Temporal Logic*, Journal of the ACM **49** (2002), pp. 672–713.

[2] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, editors, "The Description Logic Handbook: Theory, Implementation, and Applications," Cambridge University Press, 2003.

[3] Baader, F. and A. Laux, *Terminological logics with modal operators*, in: C. Mellish, editor, *14th International Joint Conference on Artificial Intelligence* (1995), pp. 808–814.

[4] Fitting, M., "Proof Methods for Modal and Intuitionistic Logics," Synthese Library **169**, D. Reidel, Dordrecht, The Netherlands, 1983.

[5] Gabbay, D., A. Kurucz, F. Wolter and M. Zakharyaschev, "Many-dimensional modal logics: theory and applications," Studies in Logic, 148, Elsevier Science, 2003.

[6] Goranko, V. and D. Shkatov, *Tableau-based decision procedures for logics of strategic ability in multi-agent systems*, CoRR **abs/0803.2306** (2008).

[7] Goré, R., *Tableau methods for modal and temporal logics*, in: M. D'Agostino, D. M. Gabbay, R. Hahnle and J. Posegga, editors, *Handbook of Tableau Methods* (1999), pp. 297–396.

[8] Hansen, H. H., "Tableau Games for Coalition Logic and Alternating-time Temporal Logic," Master's thesis, Universiteit van Amsterdam (2004).

[9] Hintikka, J., *Form and content in quantification theory*, Acta Philosophica Fennica **8** (1955), pp. 7–55.

[10] Horrocks, I., P. F. Patel-Schneider and F. van Harmelen, *From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language*, J. of Web Semantics **1** (2003), pp. 7–26.
URL download/2003/HoPH03a.pdf

[11] Horrocks, I. and U. Sattler, *A tableau decision procedure for $\mathcal{SHOIQ}$*, J. Autom. Reason. **39** (2007), pp. 249–276.

[12] Lutz, C., H. Sturm, F. Wolter and M. Zakharyaschev, *Tableaux for temporal description logic with constant domains.*, in: R. Goré, A. Leitsch and T. Nipkow, editors, *IJCAR*, Lecture Notes in Computer Science **2083** (2001), pp. 121–136.

[13] Lutz, C., H. Sturm, F. Wolter and M. Zakharyaschev, *A tableau decision algorithm for modalized $\mathcal{ALC}$ with constant domains*, Studia Logica **72** (2002), pp. 199–232.

[14] Pauly, M., "Logic for Social Software," Ph.D. thesis, University of Amsterdam (2001).

[15] Pauly, M., *A modal logic for coalitional power in games*, Journal of Logic and Computation **12** (2002), pp. 149–166.

[16] Schwendimann, S., *A new one-pass tableau calculus for pltl*, in: *TABLEAUX '98: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (1998), pp. 277–292.

[17] Seylan, I. and W. Jamroga, *Description logic for coalitions*, in: *Proceedings of AAMAS'09*, 2009, pp. 425–432.

[18] Wolter, F. and M. Zakharyaschev, *Satisfiability problem in description logics with modal operators*, in: S. S. A.G. Cohn, L. Schubert, editor, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)* (1998), pp. 512–523.